Network Coding for Efficient Broadband Data Delivery in Infrastructure-based Vehicular Networks with OpenFlow

Ke Xu[1], Saravanan Sampathkumar[1], Kuang-Ching Wang[1], and Parmesh Ramanathan[2]

[1]Department of Electrical and Computer Engineering
Clemson University
Clemson, SC, USA
{kxu, sampath, kwang}@clemson.edu

[2]Department of Electrical and Computer Engineering
University of Wisconsin – Madison
Madison, WI, USA
parmesh@ece.wisc.edu

***Abstract*—This paper proposes a system architecture for supporting efficient broadband data delivery in infrastructure-based vehicular networks. The proposed approach addresses two major challenges for high throughput data transport from Internet to moving vehicles over infrastructure wireless networks such as today's 4G technologies: 1) difficulty in maintaining stable throughput over high latency wide area paths in core networks; 2) difficulty in maintaining continuous data download across road-side units (RSUs) in the edge. Specifically, the system multicasts network-encoded packets in the core network (wired Internet infrastructure) to multiple selected RSUs, while the RSUs collaborate in disseminating and scheduling delivery of the encoded packets to vehicles. Realizing the overall system requires network coding and multipath forwarding capabilities in the core network, and network decoding support in the vehicles. For network coding to be efficient, however, dynamic control of the forwarding paths of the network coded packets are essential. This paper presents the proposed system architecture, its key components, and how they can be experimentally studied over National Science Foundation's Global Environment for Network Innovations (GENI) testbed. Experiments on ProtoGENI testbed show the feasibility and advantages of network coding in core networks.**

***Keywords*—data delivery, infrastructure, vehicular networks, heterogeneous networks, network coding, GENI**

I. Introduction

Vehicular networks are a novel class of wireless networks that have emerged as the automotive and communication industries announce plans to bring ubiquitous broadband Internet connectivity to automobiles. Envisioned applications include road safety, driver assistance, infotainment, and vehicle telematics utilizing a range of wireless communication methods such as IEEE 802.11-based standard (a/b/g/n/p/ac/ad), or 3G/4G cellular radios like WiMAX and long term evolution (LTE). Depending on the applications, such networks can be realized as an ad-hoc network, an infrastructure-based network, or a hybrid combination of the two. While vehicular ad hoc networks (VANETs) have been studied extensively by many researchers, utilizing ad hoc multi-hop communication among cars with diverse mobility pattern [1,2], industries and transportation authorities are also paying high attention to vehicle-to-infrastructure (V2I) networks due to their higher reliability and availability where such infrastructure exists. Several studies analyze and enhance the performance in V2I networks, pointing out heavy load being a primary bottleneck in large scale vehicular networks [3,4]. In [4], the author shows the system throughput quickly enters into saturation and starts to drop before the normalized traffic loads reaches 0.27.

Data delivery in vehicular networks is particularly challenging due to the high mobility, rapidly changing topology and intermittent connectivity. Vehicular relay schemes are popularly used as a way to improve the vehicular network connectivity and capacity when the channel condition is not good enough, e.g., [6]. Meanwhile, scheduling for data delivery is considered as one important and efficient mechanism to satisfy the requirements of quality of service for vehicle applications to reduce the influence by bad coverage areas, e.g., [7]. Nowadays, the availability of several access technologies such as Wi-Fi, WiMAX, LTE, offers a range of connectivity alternatives, extending the coverage of service area and minimizing the data transfer time or cost [8].

Despite the potential of heterogeneous networks to maintain more reliable link quality for connections between vehicles and infrastructures, the challenge for data delivery still exists, especially for transferring large volumes of data. The challenges come from two aspects: (1) difficulty in maintaining stable throughput over high latency wide area paths in core networks. Downloading services, such as video streaming, songs/movies/map downloads, etc., usually fetch bulk data from a remote server going through a national/international connection. Frequent handoffs across different access networks cause variations in end-to-end delays and packet drops which substantially reduced the achievable download throughput, and the retransmissions further reduce the overall network capacity. (2) difficulty in maintaining continuous data download across wireless access points (AP), base stations, or road-side units (RSUs[1]) in the edge. According to [9], at urban vehicular speeds, the median duration of connectivity between a mobile client and an AP is 13 seconds, imposing a limit on the amount of data a client can download via one AP. Once a mobile client connects with another AP, backlogged data at the previous AP can be substantial for large delay-bandwidth-product paths, and cooperation among RSUs are quite complex to set up for a

[1]The paper uses the term "RSU" to represent all types of statically deployed infrastructures in heterogeneous networks, including APs in wireless local area networks and base stations in cellular networks.

large and heterogeneous wireless infrastructure, resulting in substantial wasted bandwidth.

Network coding has been considered an effective way to make optimal use of available network resources without single point of failure by multicasting/broadcasting network encoded data blocks along multiple network paths between source(s) and destination(s). Several studies proposed inter-vehicular relaying solutions using network coding for content distribution [5, 10]. These solutions, however, do not address the challenges for continuous data download from the Internet.

To address the data delivery challenges in heterogeneous wireless edge networks and wide area core network, this paper proposes a system architecture for supporting broadband data delivery in infrastructure-based vehicular networks. The proposed system is targeted at solving two problems described above: (1) conducting network coding in the core avoids high risks of end-to-end packet losses, and (2) coordinating local RSUs to properly distribute the right sets of network coded packets to multiple RSUs based on the mobile clients' locations. OpenFlow is the underlying technology that will be used to control the forwarding paths for network-coded packets in the core networks and the edge RSUs.

The remainder of this paper is structured in the following. Section II explains the proposed system architecture and theoretical analysis. Section III introduces experiment design and experimentation plans using the Global Environment for Network Innovations (GENI) testbed, and experiment results for network coding on ProtoGENI testbed. The paper is summarized in Section IV.

II. Proposed System Architecture

This section firstly introduces the system overview and assumptions, and secondly explains the system architecture by stages. Finally, theoretical analysis is given.

A. *Overview and Assumptions*

The proposed system is designed to solve two problems of broadband data delivery: (1) stabilize high throughput and reduce end-to-end packet losses in the core over wide area paths; (2) coordinate RSUs to maintain continuous data download for vehicles. The proposed system introduces two core techniques *network coding* and *delivery scheduling* to solve these two problems. Specifically, the system multicasts network-encoded packets in the core network (wired Internet infrastructure) to multiple selected RSUs. Once the vehicle starts receiving packets through a certain RSU, actually all nearby RSUs will already be receiving or will have received encoded packets from the remote server. The nature of network coding makes such multicasting mechanism feasible and effective because, firstly the next associated RSU does not have to fetch the packets from the remote server after the vehicle enters into its cell, resulting in reducing the propagation delay and keeping high throughput use over Internet infrastructure; secondly decoding on the vehicle only requires sufficient number of encoded packets of the right set, which relaxes the requirements for upper layers to worry about out-of-order packets or dropped packets. Next, for network coding to be efficient, a local agent taking charge of a cluster of local

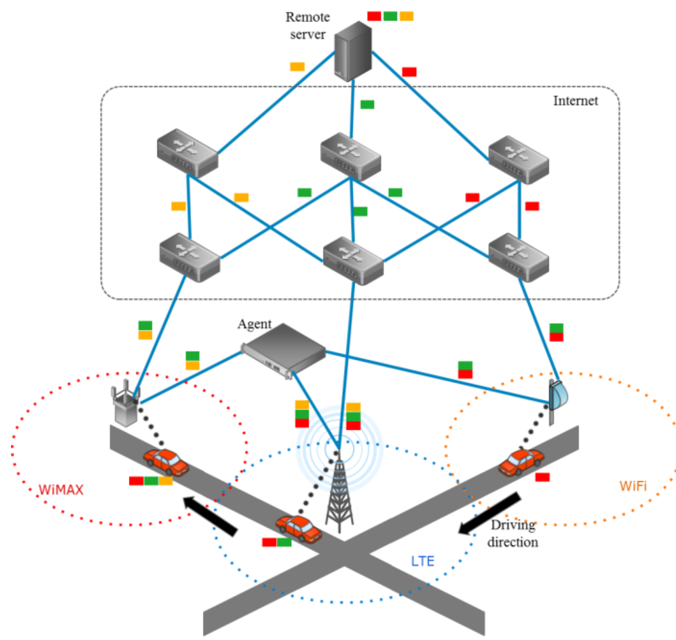

Figure 1. System architecture

RSUs is introduced to be responsible for the task of delivery scheduling. Received packets managements, next-hop RSU candidates prediction, pre-fetching notification are considered as major functions on the agent. The scheduling schemes reduce the delay and save wireless bandwidth in the edge network by predicting next-hop RSU candidates and pre-fetching mechanism.

Three major assumptions of the proposed system are given as below, (1) the vehicle is assumed to be equipped with GPS device. The device connecting to the network is assumed to be able to import data from the GPS, and integrates different wireless access technologies. The vehicle is also assumed to report its location to the associated RSU periodically, basically being 1~5 Hz based on GPS update rate. The updates will not add much traffic to the network due to the packet typically is very short (less than 100 B) [11]. (2) The agent has sufficient storage space to temporarily keep the encoded packets for RSUs, in order to reduce the heavy loads on RSUs. The agent has full knowledge of each RSU's information such as MAC/IP address, SSID, operation provider, etc. (3) The link quality is assumed to be good enough for transmitting, in other words, there is no propagation loss in the edge and no black zones. The coverage area of an RSU is assumed to be a circular cell, and the consecutive cells could be either homogeneous network or heterogeneous network. Ideal handoff is assumed when a vehicle leaves a cell.

B. *System Architecture*

Fig. 1 shows the architecture of the proposed system. We use Fig. 1 as an example to explain how the system works. The entire procedures for data delivery are separated into three stages in the following.

1) *Request stage*

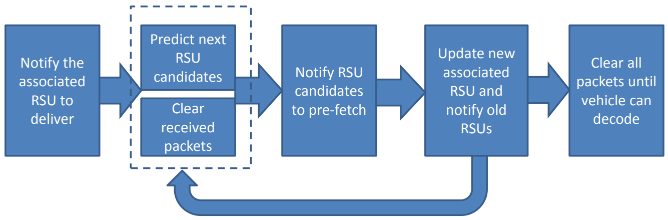

Figure 2. Delivery workflow on agent

The vehicle sends a request for data to the associated RSU, and the agent will be notified by this RSU when forwarding this request. Then the agent proceeds to notify all the local RSUs to send requests to the same remote server.

2) Multicast stage

Once the remote server receives the first request, it starts to encode the requested data using random linear network coding. The requested data will be divided into a number of elements with a certain size. The server then picks up random coefficients c_i and multiplies each element with c_i and finally adds the results, forming an encoded packet (red, green and yellow blocks shown in Fig. 1). The server randomly selects any RSUs specified by received requests to multicast the encoded packets. Note here it is not required that the server begins to encode and send only after it receives the request sent by the vehicle. Once the RSUs receive packets, they temporarily store these packets on the agent waiting for scheduling.

3) Delivery stage

The agent schedules the packet delivery following the workflow shown in Fig. 2. The agent notifies the current associated RSU to fetch and deliver its received packets. Once the delivered packets are successfully received by the vehicle, the agent clears these packets in storage voiding redundant delivery. At the same time, it predicts the future location of the vehicle based on historical GPS records, and vehicle's driving information such as speed. By predicting, the agent notifies all RSU candidates, who potentially could be the next associated RSU for this vehicle, to pre-fetch the pending packets stored on the agent (it is possible to pre-fetch pending packets received by other RSUs if its own is not sufficient). Once the vehicle executes handoff and the new associated RSU will notify the agent, and the old RSUs including those candidates are not chosen by the vehicle will be notified by the agent to clear all buffered packets. The agent loops the prediction step following Fig. 2 until the vehicle is able to decode all received packets to obtain the original message, and finally the agent clears all stored packets. In Fig. 1, the vehicle receives red, green, and yellow packet in WiFi's, LTS's, and WiMAX's coverage, respectively. The latter two delivered packets are pre-fetched by RSUs.

C. Analysis

We abstractly analyze the performance of the proposed system. All RSUs are assumed to have a uniform communication range R, and vehicle is assumed to have a constant velocity v. Vehicle traverses a cell going through the diameter, so the traverse time in one cell is $\tau=2R/v$. The remote server is assumed to reach all RSUs through a uniform number of hops n_{hop}; all one-hop link (either wired or wireless) is assumed to have a constant propagation delay d; and n_{rsu} is the number of RSUs travelled through until the vehicle is able to decode. It is not hard to derive the delay of the proposed system and conventional routing method:

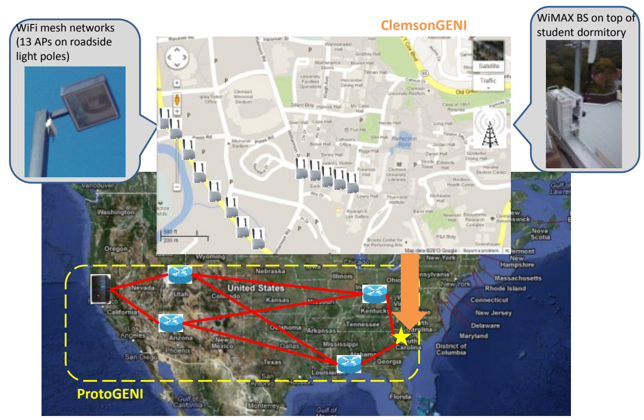

Figure 3. Experiment plans

$$D_{proposed} = O(n_{rsu}) + O(n_{hop}) \qquad (1)$$

$$D_{routing} = O(n_{rsu} \times n_{hop}) \qquad (2)$$

Compared (1) with (2), it is easily to see the delay will become very large for conventional way if either one or both of n_{rsu} and n_{hop} is large. The velocity v is directly proportional to n_{rsu}, which means the faster the velocity is, the delay performance of conventional routing is worse. In fact, n_{rsu} is different between (1) and (2) in real scenarios because using pre-fetching mechanism the vehicle will travel through less number of RSUs than conventional routing until it is able to decode.

III. Experiments using GENI Testbed

This section introduces the experimentation plans using GENI testbed, including core network and edge network. Then preliminary simulation and future work are given.

A. Experiments Plan using GENI testbed

The system is to be tested by conducting experiments using GENI testbed. GENI offers a national facility that supports exploration designs for a future global networking infrastructure containing both wired and wireless testbeds[15]. As shown in Fig. 3, we plan to combine ProtoGENI [16] and Clemson GENI Wireless testbeds to realize the proposed system, which is described in the following.

1) Implementing network coding using ProtoGENI

The network coding part will be tested across the country by reserving several geographically distributed nodes on ProtoGENI. ProtoGENI backbone is built on top of the Internet 2 infrastructure, using HP switches to provide VLAN connectivity across US. Designing topologies by reserving backbone nodes (each backbone node includes 2 Dell 2950 PCs having at least one 1Gbps Ethernet interface connecting to the switch) at different sites are appropriate to be a national

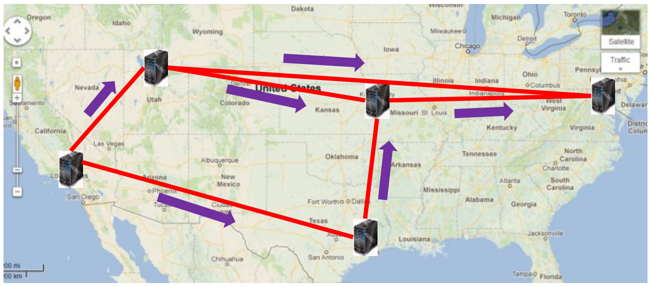

Figure 4. Network coding experiments on ProtoGENI testbed

connection in the core network running network coding. Flack tool will be used to create slices and slivers, reserve and manage resources as well. The network coding code will be executed on the remote server, and the vehicle's device. Other nodes only function as forwarding nodes. An OpenFlow controller can be used to control the forwarding paths of encoded packets.

2) Implementing delivery scheduling over Clemson Wi-Fi and WiMAX network

Among one of the major campuses involving in GENI projects, Clemson developed and deployed an outdoor Wi-Fi mesh network and WiMAX network as a part of GENI wireless infrastructures. There are 13 Wi-Fi APs (PC Engines-based device using IEEE 802.11b/g protocol) installed on road-side light poles and one WiMAX base station (Airspan) installed on the roof of the highest student dormitory in the campus, as marked in Fig. 3. Although recent high data rate Wi-Fi variant (IEEE 802.11ad could achieve 7Gbps for data rate) is more suitable for high throughput use, the proposed system architecture is not limited to deployed Wi-Fi networks. The experiments will be designed properly such that results could be interpreted proportionally based on the available bandwidth. More importantly, the delivery scheduling problem is the focus in this system.

3) OpenFlow roles

OpenFlow is the underlying technology that will be used to control the forwarding paths for network-coded packets in the core networks and the edge RSUs. The requirement for using OpenFlow in core networks is existence of OpenFlow-enabled switch being involved in multicast. A remote controller could be used to control the forwarding paths for each hop in core networks. Further implementation and studies could consider optimizing the forwarding path selection based on certain metrics, such as traffic loads.

For edge networks, different types of wireless infrastructures are connected to OpenFlow enabled HP switches through VLAN connectivity. OpenFlow provides the possibility and convenience for centralized controlling in the edge networks. A private VLAN at Clemson could be created and attached to all the switch ports connecting wireless infrastructures. A server on which the agent functions perform will also be attached on this VLAN. A single Floodlight controller [14] is used to control the edge network. Due to the HP switches is on GENI VLAN over Internet 2, the core network over ProtoGENI and the edge network over Clemson OpenFlow network is able to combine together.

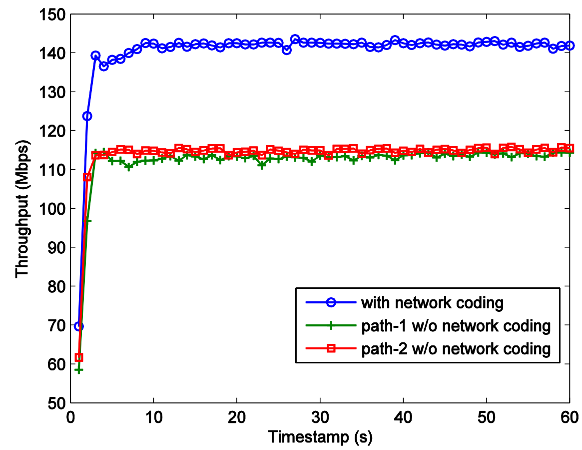

Figure 5. End-to-end throughput

B. *Experiments Results and Setup on ProtoGENI testbed*

Experiments of network coding were conducted on ProtoGENI tested. We implemented network coding in Java based on open-source ncutils library [12], using random network coding described in Sec.II.B. The network topology is shown in Fig. 4, in which five ProtoGENI nodes were reserved, being located in Los Angeles (CA), Salt Lake City (UT), Houston (TX), Kansas City (KS), and Washington DC (DC), respectively. The red links in Fig. 4 are the LANs created on ProtoGENI testbed. The node in Los Angeles is the server and the node in DC is the client who fetches the data from the server. The remote server divides the request message into 10 elements and sends the encoded packets in round robin order to the next hops. The nodes who receive the encoded packets forward them to all their available next hops. The purple arrows in Fig. 4 are the traffic directions on all links.

End-to-end throughput is measured in the designed network, compared to unicast cases. Fig. 5 shows the throughput performance for one run with network coding and unicast on two different paths without network coding. Path-1 follows the route CA-UT-DC, and path-2 follows CA-TX-KS-DC. Five runs were tested and the throughput performance was found stable. Fig. 5 indicates that throughput performance on both unicast paths are similar but with network coding the throughput increases by about 25~30%. In order to examine how much bandwidth cost in core network is added for getting the throughput benefit, the results by averaging five runs are shown in Table I. More interestingly, the bandwidth cost with network coding is not the most expensive among three cases. The required bandwidth on path 2 is even much larger than that with network coding due to 3 hops are involved in unicast, with consuming large bandwidth on each link. From Fig. 5 and Table I, network coding is beneficial to throughput performance improvement but requiring comparative bandwidth cost with unicast in core networks. The traffic on each link is more distributed so that the risks of end-to-end packet losses are reduced and the requirement of traffic load on each link is relaxed as well.

Table I. Bandwidth cost in core networks

Bandwidth /Mbps	CA-UT	CA-TX	UT-DC	UT-KS	TX-KS	KS-DC	Total cost	Throughput	Efficiency
Network coding	72.6	72.6	71.6	71.6	72.1	140.6	501.1	141.5	0.28
Path-1	211.9	N/A	211.9	N/A	N/A	N/A	423.8	111.2	0.26
Path-2	N/A	212.5	N/A	N/A	212.5	212.5	637.5	113.8	0.18

The experiment setup on ProtoGENI testbed is introduced by steps as below.

Step-1 Resource reservation: Register a new slice using Flack. On the slice create a sliver specifying nodes (pg56, pg45, pg55, pg42, pg40) and defining network topology. Submit the request and wait for getting and redeeming the resource ticket.

Step-2 Operating system (OS) loading and experiment preparation: The OS image is specified when creating the sliver, and fedora 10 standard image was used in our experiments (it is also supported to directly load customized OS image). Then update and install any necessary software and dependencies. Java-1.6.0-openjdk.i386 is downloaded and installed in the experiments. Download, compile or/and install any user application. We setup a remote ftp server where network coding application is downloaded.

Step-3 Run experiments and collect raw data: Login every node to launch the experiments. After finishing save the raw data.

Step-4 Cleanup: Delete the sliver using Flack. The slice will be automatically deleted after some hours.

IV. Summary and Future Work

This paper proposes a system structure supporting for efficient broadband data delivery in infrastructure-based vehicular networks. Network coding is used to multicast encoded message packets to all local RSUs in the core network. In the edge the local agent manages received packets, predicts the next RSU candidates and schedules the delivery. Theoretical analysis shows the feasibility of network coding and advantage of reducing delay. Experiments over GENI infrastructures are planned. Results from network coding experiments on ProtoGENI testbed indicate that the end-to-end throughput is improved by 25~30% but not requiring more bandwidth cost in core networks. The experiment setup is also introduced and summarized for reference by other experimenters.

Future work is the development and implementation of the agent, whose main functions and operation logic follow the delivery workflow depicted in Fig. 2. The experiments will be designed and conducted combining both GENI wired and wireless testbeds finally.

Acknowledgment

The authors are supported in part by National Science Foundation grant CNS-1049947, CNS-1060344 and GENI subcontracts1833A and 1843C.

References

[1] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta, "A cooperative approach to traffic congestion detection with complex event processing and VANET." *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 914-929, 2012.

[2] L. Cheng, and S. Panichpapiboon, "Effects of intervehicle spacing distributions on connectivity of VANET: a case study from measured highway traffic." *IEEE Commun. Mag.*, vol. 50, no. 10, pp. 90-97, 2012.

[3] K. Xu, B. T. Garrison, and K.-C. Wang, "Performance modeling for IEEE 802.11 vehicle-to-infrastructure networks with directional antennas," in *Proc. IEEE VNC 2010*, Jersey City, NJ, USA, 2010, pp. 215-222.

[4] K. Xu, B. T. Garrison, and K.-C. Wang, "Throughput modeling for multi-rate IEEE 802.11 vehicle-to-infrastructure networks with asymmetric traffic." in *Proc. ACM MSWiM 2011*, Miami Beach, FL, USA, 2011, pp. 299-306.

[5] A. Khlass, Y. Ghamri-Doudane, and H. Gacanin, "Combining cooperative relaying and analog network coding to improve network connectivity and capacity in vehicular networks," in *Proc. IEEE GLOBECOM 2011*, Houston, TX, USA, 2011, pp. 1-5.

[6] S. C. Ng, W. Zhang, Y. Zhang, Y. Yang, and G. Mao, "Analysis of access and connectivity probabilities in vehicular relay networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 140-150, 2011.

[7] K. Xu, P. V. Orlik, Y. Nagai, and M. Saito, "Location based data delivery schedulers for vehicle telematics applications," in *Proc. IEEE VTC 2012 Fall*, Québec City, Canada, 2012, pp. 1-5.

[8] K. Shafiee, A. Attar, and V. C. Leung, "Optimal distributed vertical handoff strategies in vehicular heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 3, pp. 534-544, 2011.

[9] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *Proc. ACM MobiCom* 2006, Los Angeles, CA, US, 2006, pp. 50-61.

[10] S. Ahmed, and S. S. Kanhere, "VANETCODE: network coding to enhance cooperative downloading in vehicular ad-hoc networks," in *Proc. ACM IWCMC 2006*, Vancouver, British Columbia, Canada, 2006, pp. 527-532.

[11] CAMP VSC, "Vehicle Safety Communications Project Task 3 Final Report - Identify Intelligent Vehicle Safety Applications Enabled by DSRC," USDOT, 2005.

[12] Ncutils. Network Coding Utilities. [online]. 2012, http://code.google.com/p/ncutils/ (Accessed: 25 Jan 2013)

[13] Mininet. [online] . 2013, http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/Mininet (Accessed: 25 Jan 2013)

[14] Floodlight Openflow Controller. [online]. 2013, http://floodlight.openflowhub.org/ (Accessed: 25 Jan 2013)

[15] GENI. [online]. 2013, http://www.geni.net/ (Accessed: 25 Jan 2013)

[16] ProtoGENI. [online]. 2013, http://www.protogeni.net/ (Accessed: 25 Jan 2013)